**École internationale des sciences du raitement de l'information**

****

**Master of Engineering – Meng -** **Big Data**

**Heuristics**

**Maximization of a Building Surface on**

**a Parcel of Land**

Aytaç Özkan

**2019, Cergy**

**1. Introduction**

Regarding the architectural institution proposed the problem that we try to optimize the maximization of a building’s surface on a parcel of land with heuristic and stochastic optimization methodologies and algorithms.

**2. Problem Implementation on Python**

**2.1. Particle Swarm Optimization (PSO)**

Solves a problem by having a population (in our case polygons, rectangles) of candidate solutions, particles and moving these particles around in the search-space according to simple mathematical formulae and algorithms over the particles’ position and velocity.

Each particles’ movement is influenced by its local best known position, but it also guided toward the best known positions in the search-space which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

**2.1.1 Algorithm**

**for each particle i = 1, ..., S do**

Initialize the particle's position with a uniformly distributed random vector: xi ~ U(blo, bup)

Initialize the particle's best known position to its initial position: pi ← xi

**if f(pi) < f(g) then**

update the swarm's best known position: g ← pi

Initialize the particle's velocity: vi ~ U(-|bup-blo|, |bup-blo|)

**while a termination criterion is not met do:**

**for each particle i = 1, ..., S do**

**for each dimension d = 1, ..., n do**

Pick random numbers: rp, rg ~ U(0,1)

Update the particle's velocity: vi,d ← ω vi,d + φp rp (pi,d-xi,d) + φg rg (gd-xi,d)

Update the particle's position: **xi ← xi + vi**

**if f(xi) < f(pi) then**

Update the particle's best known position: pi ← xi

**if f(pi) < f(g) then**

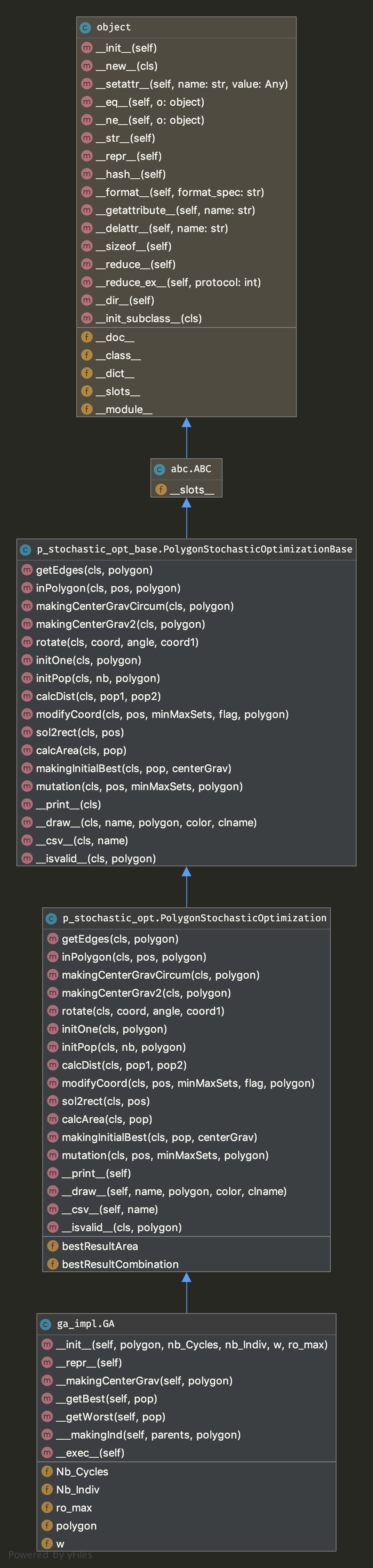
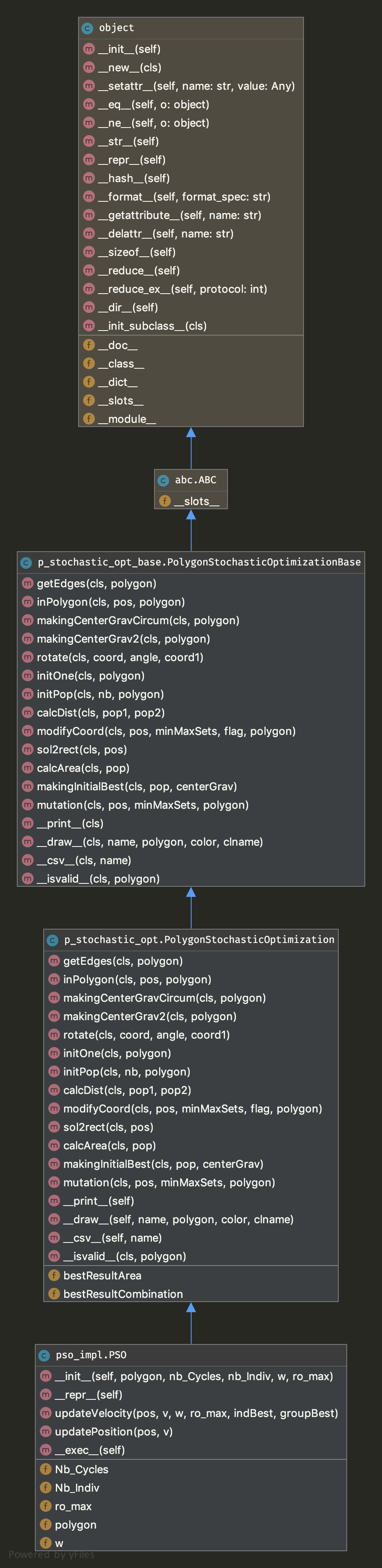
Update the swarm's best known position: g ← pi

**2.2 Genetic Algorithms**

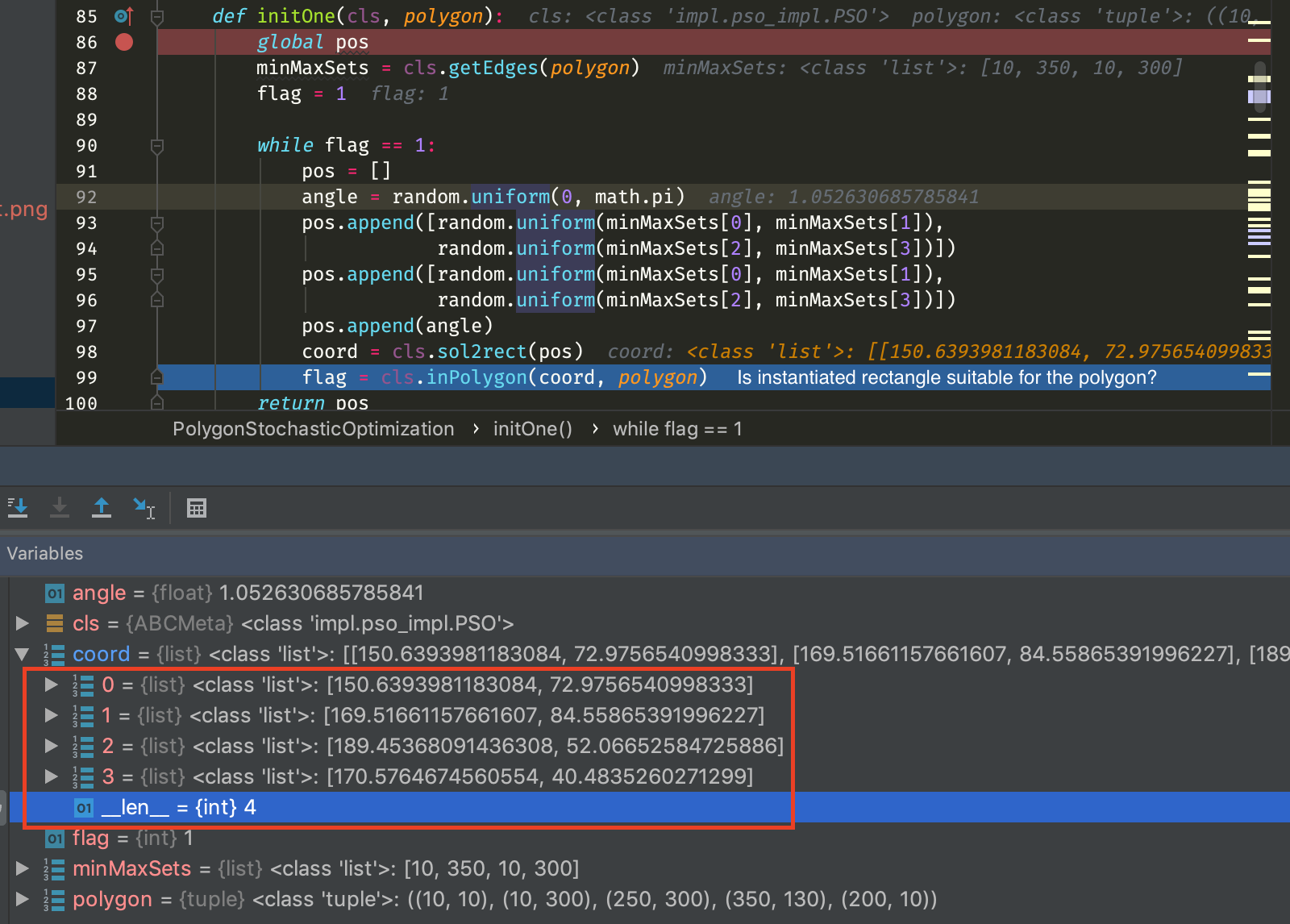
In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

**2.3. Application Structure**

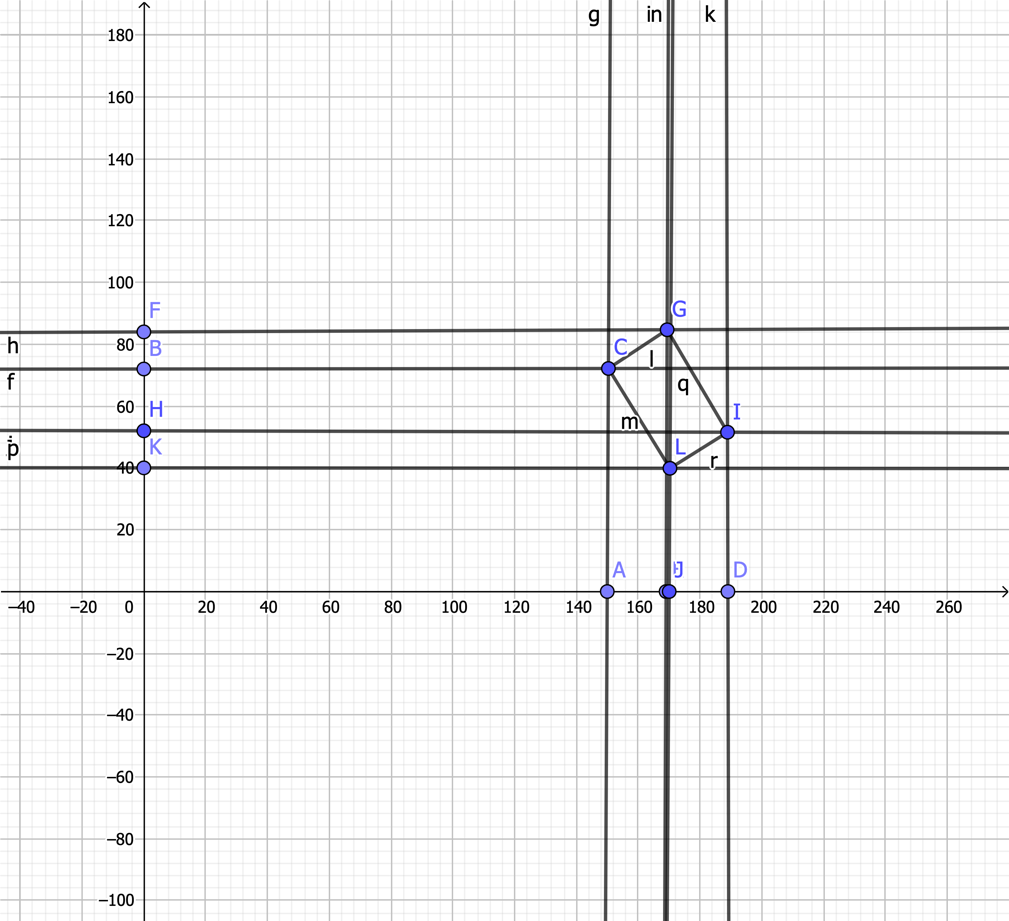
As we explained at previous topics, we implemented Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to find the best a building’s surface on a parcel of land with python programming language. Respectively GA and PSO class as below:

As you see ‘initOne’ function has initialized a rectangle in order to fi for polygons given by us.

****

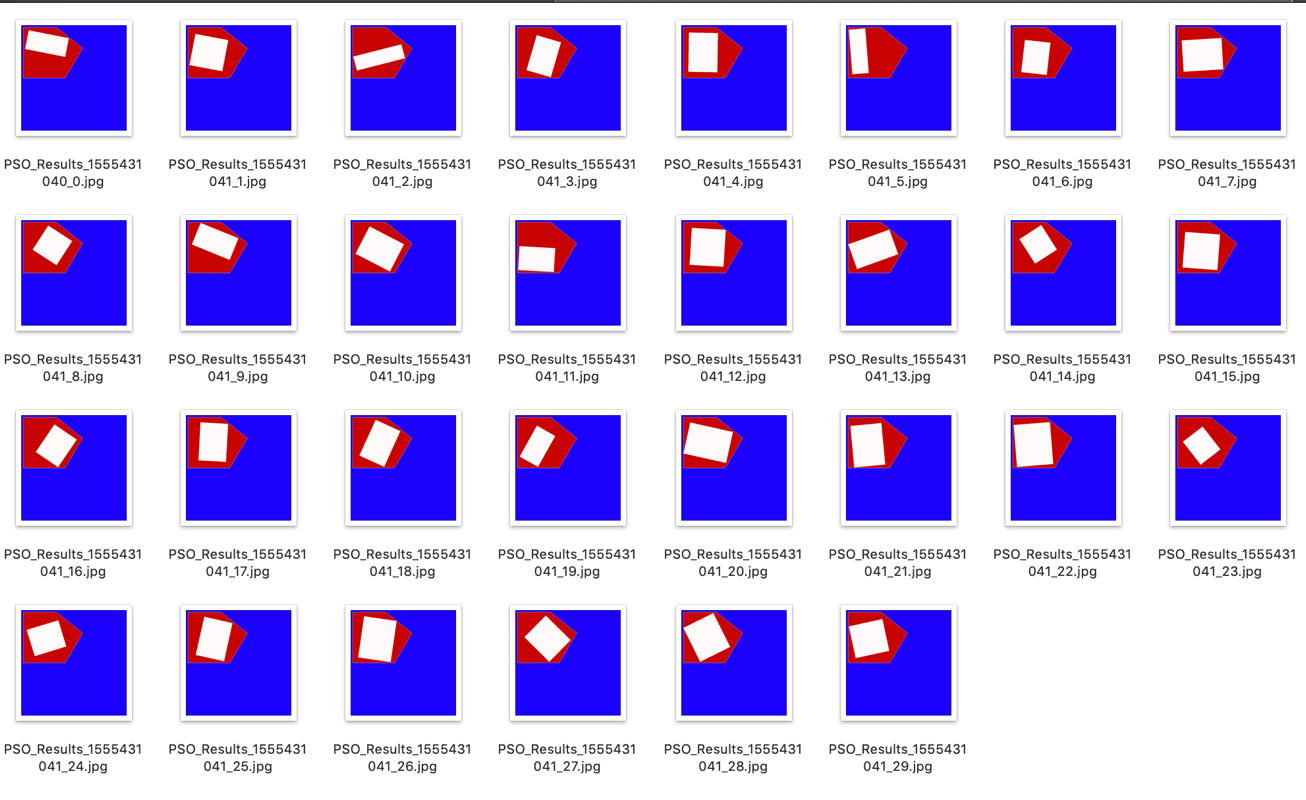
Result of initOne’ function after execution.

****

**3. Results and Output**

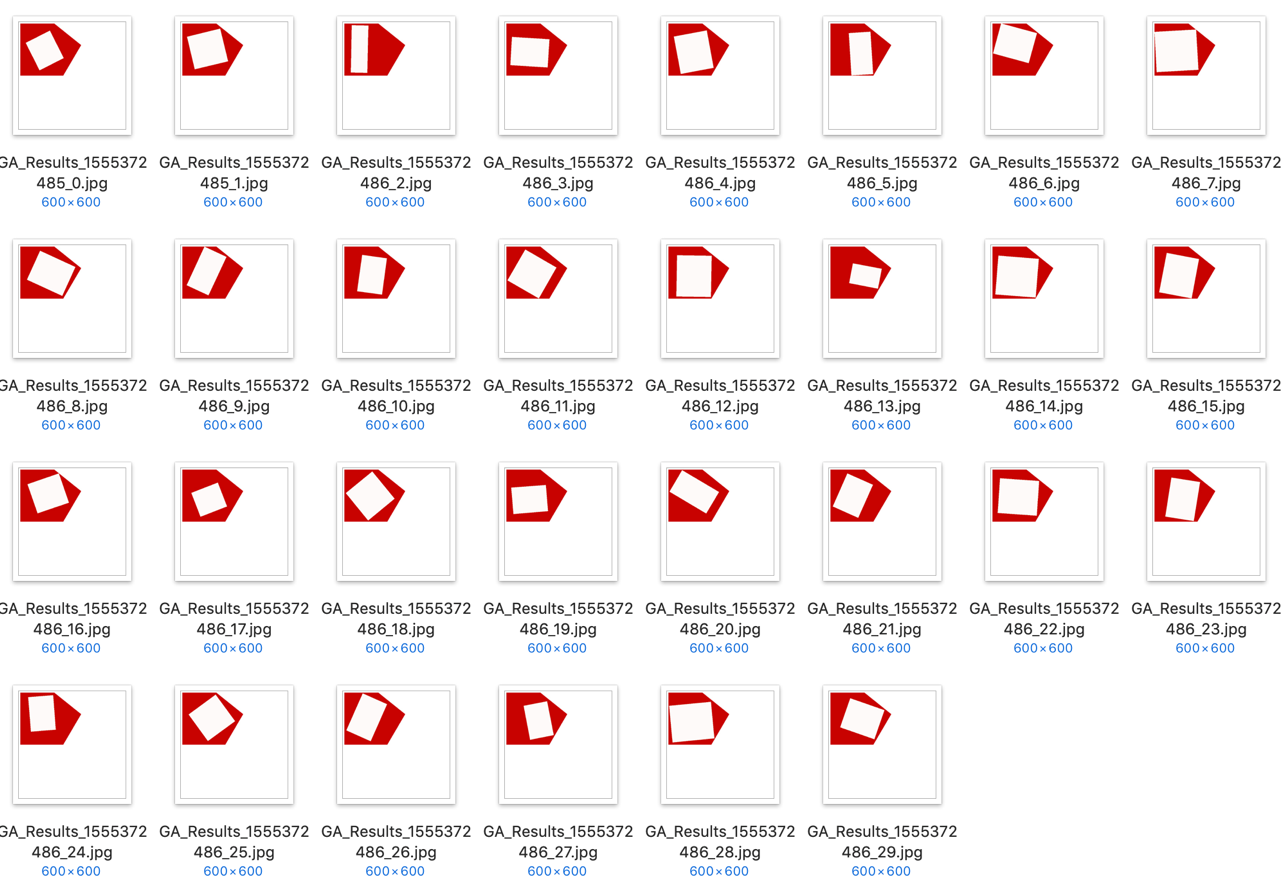
PSO algorithm is results as below also you can find them on project folder under ‘images folder.’

Generated images:

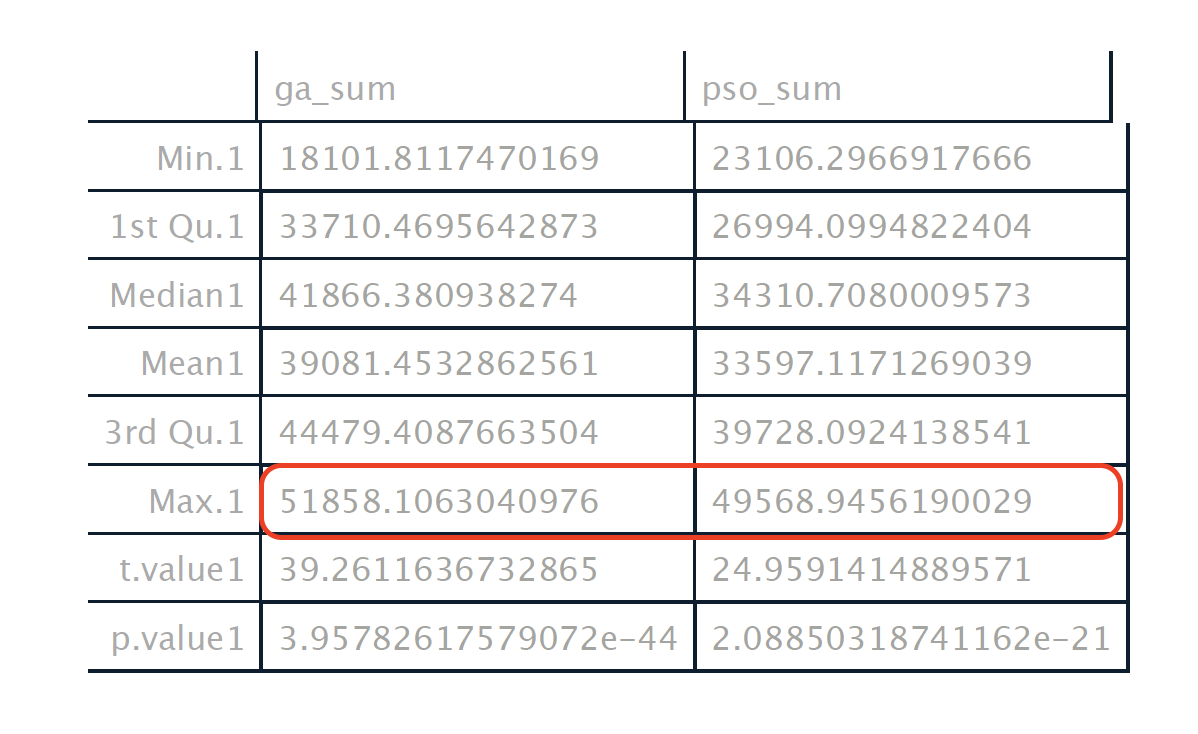


GA algorithm is results as below also you can find them on project folder under ‘images folder.’

Generated images:



On the other hand after execution of both algorithms, we generated csv files and save the best result areas. Then we evaluated statically our results with R programming language. Interested \*r file can be found in the project work dir.



Regarding to our results GA algorithm gives more accurate **t (“39.2611636732865”)** and **p** **(“3.95782617579072e”)** value according to PSO algorithm. with 95 percent confidence interval. Also you can find the generated data set in the project working dir.